

# Energy-Constrained UAV Flight Scheduling for IoT Data Collection with 60 GHz Communication

Wenjia Wu, Shengyu Sun, Feng Shan, Ming Yang, and Junzhou Luo

**Abstract**—In recent years, the data from Internet of Things (IoT) devices is growing at a rapid pace, and the data collection issues have attracted more and more attention. Distinct from existing solutions which usually adopted traditional wireless technologies achieving low-bandwidth data connections towards IoT devices, this paper adopts the 60 GHz communication technology which is with increasing maturity, providing high-bandwidth wireless transmission for data-intensive IoT devices, such as high-definition (HD) cameras. However, 60 GHz links are subject to line-of-sight short communication range, therefore, we propose to use unmanned aerial vehicles (UAVs) to fly over data-intensive IoT devices and achieve short-range high-throughput 60 GHz transmission in this paper. Moreover, for a set of HD cameras deployed in the linear scenario, multiple UAV flights are assigned to collect data by 60 GHz communication. To this end, we investigate the UAV flight scheduling (UFS) problem which aims to minimize the number of UAV flights while satisfying the data requirements of all ground cameras (GCs) with limitations of UAV's energy and data storage. We prove that the UFS problem is NP-hard and design efficient algorithms with constant theoretical approximation ratios. Specifically, we first study a special case of the UFS problem where all the cameras are on the same direction of the UAV ground station, and propose two algorithms NF\_SUFS and FF\_SUFS, whose approximation ratios are both proved to be 2 by theoretical analysis. Then, we extend the algorithms to a more general case with the cameras on both directions of the ground station along the road, and put forward the FF\_UFS algorithm that achieves an approximation ratio of 3. Finally, we conduct experiments to validate the effectiveness and efficiency of our algorithms.

**Index Terms**—UAV-aided data collection, 60 GHz communication, flight scheduling, energy constraint.

## I. INTRODUCTION

The Internet of Things (IoT) is a popular network paradigm where massive number of objects, such as sensors, RFID tags, cameras, smart phones, computers and people, are able to interact with each other and achieve common goals [1]. In recent years, IoT has been widely applied in various applications, such as smart city [2], smart agriculture [3], and industrial systems [4].

With the development of IoT, a large number of heterogeneous devices and numerous applications are emerging, and thus multiple wireless technologies are needed to be adapted to different types of scenarios. As one of the most promising

wireless technologies, 60 GHz communication provides multi-Gbps links that can be well applied in wireless high-definition (HD) multimedia, virtual reality and so on [5]–[7]. Its powerful advantage of high bandwidth enhances the ability of data transmission for some data-intensive IoT devices, such as HD cameras.

In urban environment, some infrastructures, such as roads, bridges, rivers, pipelines, and fences, need cameras deployed for video surveillance of safety prevention or environmental monitoring, which can be called linear scenarios. The cameras distributed linearly along the infrastructure can capture the events and store the data in their own storage. The data must be transmitted to remote servers for long term storage, effective data management and analysis. The video data generated by the camera can be transmitted through a wired way or wirelessly. When the network infrastructures such as base stations and cables are inadequate or destroyed, or when temporary monitoring is needed, the wireless data transmission is preferred. In previous work, some researchers have considered collecting data via wireless networks [8], [9], where each ground node delivers its stored data to a sink node via multi-hop wireless transmission. However, the solutions offer low-rate data transmission which is not suitable for efficient data collection of data-intensive nodes. As 60 GHz technology continues to develop, it has shown remarkable effect on transmitting high bandwidth data. The wireless network employing 60 GHz communication is flexible in deployment, convenient in network distribution, and able to transmit high bandwidth video data in a short time, but it also brings new problems. 60 GHz millimeter waves decay rapidly in air and are difficult to be used for long-distance communication [10]–[12]. When deploying wireless networks to collect data, the distance between nodes is limited by the communication distance, and the introduction of 60 GHz exacerbates this limitation. Besides, multi-hop transmission in linear scenarios will undoubtedly increase the energy consumption of relay nodes and shorten the network lifetime. In order to solve this issue, mobile sinks can be used to approach the IoT nodes and receive data [4], [13]. In this way, not only the flexible deployment of the network can be realized, but also the appropriate type of mobile sinks and data communication protocols can be selected to adapt to various IoT networks. Moreover, mobile sinks can well make up for the disadvantage of short-distance 60 GHz communication.

Due to the advantages of mobility, flexibility and adaptive altitude, unmanned aerial vehicles (UAVs) have attracted extensive attention in academic and industry [14]–[16]. For example, UAVs equipped with wireless communication modules

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

W. Wu, S. Sun, F. Shan, M. Yang, and J. Luo are with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: wjwu@seu.edu.cn, shengyusun@seu.edu.cn, shanfeng@seu.edu.cn, yangming2002@seu.edu.cn, jluo@seu.edu.cn).

Corresponding author: Wenjia Wu

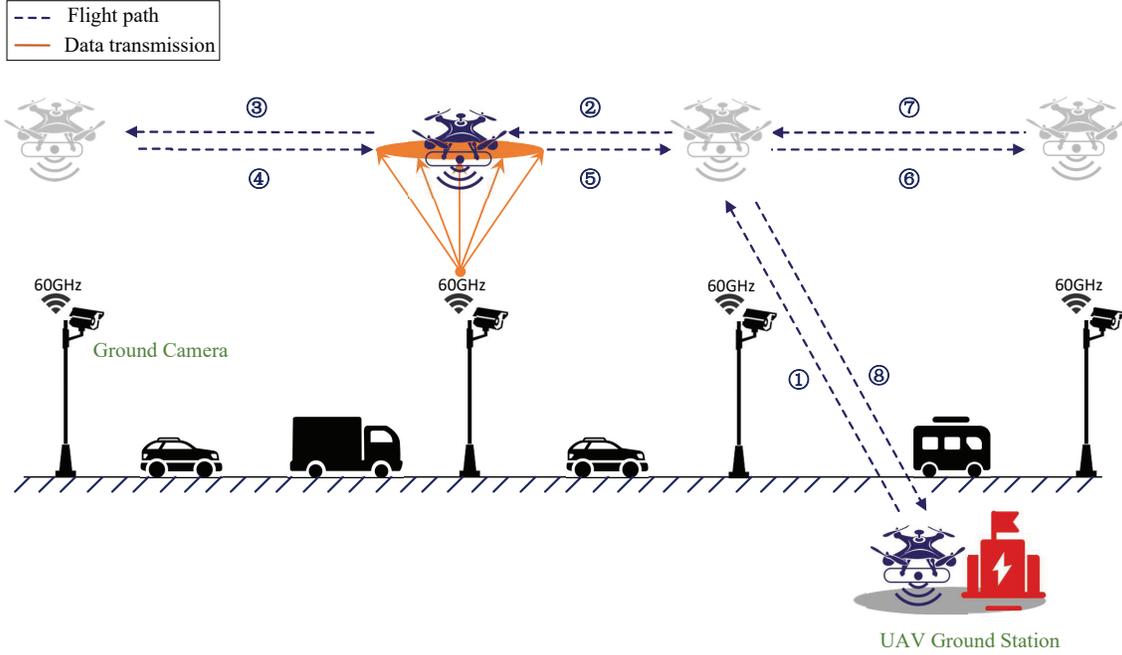


Fig. 1. A UAV-aided data collection scenario with 60 GHz communication. The UAV starts from the ground station, flies along the road, and chooses the appropriate ground cameras on the path to collect data in flying and hovering mode. The UAV collects data no more than its storage, and must fly back to the ground station before running out of its energy.

can deliver reliable, cost-effective, and on-demand wireless services to desired areas, and they can also wirelessly charge sensors to provide them with energy for data transmission [17]. In recent years, some researchers have begun to use a UAV as mobile sink for such data collection, where the ground node can directly send its data to the UAV and the UAV will bring the data to remote servers by flying. Because the UAV can move very close to the ground node before transmission, and the line-of-sight link of ground-to-air transmission is better than that of ground-to-ground transmission. In addition, compared with other types of mobile sinks (e.g., vehicles), the UAV flies in the air and does not depend on the road conditions on the ground. Since UAVs are energy constrained due to the limited on-board battery, the energy-efficient UAV-aided data collection has become a growing research interest.

In the UAV system, there is a UAV ground station as the control center. The UAV starts from the ground station, collects data and returns to the ground station. In this paper, this round trip is called as a *UAV flight*. Because the resources of UAV are limited, such as energy and data storage, multiple UAV flights are needed for the data collection of data-intensive IoT network. However, most of existing research works only consider the scenarios with a single UAV [18]–[21], and cannot solve the issues in the scenarios that require multiple UAVs or multiple UAV flights. Since each UAV flight requires certain costs, such as energy costs, maintenance costs and hardware costs, it is a significant and challenging issue to minimize the number of UAV flights while completing data collection tasks. Sometimes when the amount of data is large, the UAV needs to hover at a certain position for data collection. Thus, the UAV's energy is in general composed of two aspects, i.e.,

the propulsion energy consumption for flying and hovering, and the communication related energy consumption. In linear scenarios, the farther a UAV travels, the more nodes it can choose to serve and the more energy consumed for flying, but the less energy available for hovering and communicating with ground nodes, and vice versa. A trade-off should be made between the energy consumption for flying to target nodes and the energy consumption for hovering to collect data (including hovering and data transmission). If the UAV flights are not well scheduled, a large amount of energy will be spent on unnecessary flying, which will reduce the energy available for data collection and ultimately lead to the increase of flights and costs. Further, when the ground station is located at arbitrary position, the UAV needs to choose the ground nodes of which direction to serve, making the problem more complicated and difficult to be solved.

In this paper, we consider the scenario that UAVs need to collect video data from a set of ground cameras along a road, where each camera transmits data to the UAV by 60 GHz communication, and the UAV brings data to the ground station, as shown in Fig. 1. In such a scenario, the UAV collects data from cameras in flying and hovering mode. Due to the limitation of the UAV's energy and data storage, multiple UAV flights may be essential to serve all the cameras. Hence, we study the UAV flight scheduling (UFS) problem with the objective of minimizing the number of UAV flights while all the cameras can be served.

The contributions of this paper are summarized as follows:

- We consider multiple UAV flights for 60 GHz data collection in linear scenarios, and formulate the UFS problem as a mixed integer linear programming (MILP) model,

which aims to minimize the number of UAV flights while satisfying data requirements of ground cameras with the limitations of energy and data storage. In addition, we also prove that the UFS problem is NP-hard.

- We study a special case of the UFS problem where all the cameras are on the same direction of the ground station, and propose the NF\_SUFS and FF\_SUFS algorithms. According to the optimal solution of the relaxed problem, we prove that the approximation ratio of our proposed algorithms is 2.
- We extend the algorithms to a more general case with the ground station located at an arbitrary position, and put forward the FF\_UFS algorithm that assigns UAV flights for the data collection on two directions. We explore that the sum of the optimal solutions for two single-direction subproblems is less than 1.5 times the optimal solution of original problem, and prove that the FF\_UFS algorithm achieves an approximation ratio of 3.
- We implement the proposed algorithms and conduct simulations to evaluate the performance of our proposed algorithms. Simulation results demonstrate that the average performance of FF\_UFS algorithm is at most 4.8% higher than the near optimal solutions of the MILP model, which is better than the theoretical approximation ratio.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents the system model and formulates the problem. Section IV designs approximation algorithms for the single-direction UFS problem. On this basis, Section V studies the general case where the ground station is located at arbitrary position. Numerical results are presented in Section VI. We conclude the paper in Section VII.

## II. RELATED WORK

UAV flight scheduling for IoT data collection has become a interesting research issue, and there are several works to study this issue in the past few years. In the following, we discuss the existing solutions in two different kinds of scenarios, i.e., two-dimensional scenarios and linear scenarios, and then introduce the flying mode and hovering mode for UAV-aided data collection.

### A. Two-dimensional/Linear Scenarios of IoT Data Collection

Generally, the researchers considered IoT networks in a two-dimensional area. In the early IoT networks, sensor nodes were densely deployed in certain scale ranges and data collection could be achieved through multi-hop wireless links, with the aim of maximizing the data collecting rate [8], [22], improving energy utilization [9], and so on. However, there still exist some issues, such as low energy efficiency and limited communication distance. To break above limitations, researchers found the importance of mobility for IoT networks [23], [24] and investigated the data collection problem with mobile elements (MEs) [25], [26]. They studied the role of MEs and categorized MEs according to the tasks they need to perform. On this basis, Gao et al. [27] studied the problem of efficient data collection in IoT networks with the constraints that the mobile sink can only move periodically

along a fixed path at a certain speed. They proposed a heuristic based on genetic algorithm and local search to maximize the amount of data collected by mobile sinks and balance the energy consumption. Similarly, Abdulla et al. [28] considered a UAV glided around the sensor field in a circular trajectory. They equipped the cluster heads with transceivers capable of adaptive modulation, and adopted potential games to allocate the UAV's time slots among cluster heads in order to improve network utility while achieving fairness among cluster heads. However, they have not considered the influence of ME's traveling path on the objective optimization. Subsequently, some researchers have begun to emphasize the impact of path planning and jointly consider task scheduling on the research problem [29]–[31]. To be specific, Tuyishimire et al. [29] proposed a persistent path planning and UAV scheduling model, using a group of UAVs from different ground stations to complete the data collection task of ground sensors and send the collected information to the nearest ground station, with the aim of minimizing the total energy consumed by all UAVs. In particular, they noted the fact that different sensors might need to be served at distinct expected time and proposed the corresponding heuristic algorithm. Resembly, Zhang et al. [30] studied the problem of how to schedule the minimum number of charging vehicles to charge lifetime-critical sensors, and proposed algorithms to decompose the traveling salesman problem (TSP) path according to the battery capacity of each vehicle. Furthermore, they considered UAV-aided data collection scenarios, and introduced the concept of edge weight threshold. On this basis, the improved algorithm called *approAlgNoNei* is proposed to ensure the freshness of the collected data [32].

However, if each sensor node needs to be served once and only once, these solutions could not perform well. For multi-UAV-aided data collection in two-dimensional scenarios, the UAV scheduling problem is often seen as a variation of TSP, and the corresponding solutions can be divided into two categories. First one is to divide the target region and then arrange a UAV for each sub-region and carry out path planning [33]. For each sub-region, the researcher can regard the problem as a TSP with the goal of minimizing UAV's energy consumption and propose heuristic algorithms. Although the optimization of energy efficiency is considered, the total energy consumption of a UAV is not limited. This may lead to a special case, i.e., the total energy consumption exceeds the capacity of UAV battery, which makes these solutions impractical. The second one is to regard the problem as a variant of TSP first and then divide the generated path according to UAV resource limitations and data requirements [30], [32]. Since the TSP is an NP-hard problem, heuristic algorithms are proposed when the scale of the node is large. The segmentation based on the non-optimal solutions of TSP may lead to unnecessary UAV flights, and significantly affect the performance of these algorithms.

Nevertheless, in some scenarios, sensors may be arranged in sequence, in order to monitor some special structures with linear nature such as oil/gas pipelines, roads, bridges, rivers, and coasts. In theory, this linear scenario is a special case of two-dimensional scenarios, so all the above solutions can also

be applied to linear scenarios. In the linear scenario, since the path to a specific node is unique, intermediate nodes along the path are also determined. Therefore, when the farthest served node is determined, the flying energy of the whole UAV flight can be known in advance, and the selection of the intermediate nodes along the path do not need to consider energy consumption for flying. If the algorithms designed for two-dimensional scenarios are applied in linear scenarios, the above characteristics of linear scenarios are ignored, and thus their performance could not be well.

Compared with two-dimensional scenarios, there are relatively few studies specific to linear scenarios. Existing research works mainly focus on the parameter control of UAVs, the selection of cluster heads, and the data collection mode. Jawhar et al. [18] presented a framework for monitoring linear infrastructures using UAVs. They proposed three different UAV movement approaches, and measured the system's delivery ratio and average delay under various network conditions. Similarly, Ren et al. [19] studied the data collection aided with a mobile sink in an energy harvesting sensor network for traffic monitoring and surveillance purpose on busy highways. They first formulated a data collection maximization problem that dealt with multi-rate transmission mechanism and transmission time slot scheduling among the sensors. By contrast, Gong et al. [20] studied the scenario where a UAV collected data from a set of sensors on a straight line and satisfied the data requirements of all sensors. They jointly optimized flight speed, transmission power and intervals to minimize total collection time. However, they did not take into account of the system's energy consumption. Vishnuvarthan et al. [21] concentrated on the speed of mobile data collector and each cluster's data transmission range to solve the energy hole problem and network lifetime reduction in strip-based network. They only considered the energy consumption of sensors, while ignoring that the energy of the mobile collector was also limited and its tasks needed to be properly planned. As discussed above, most existing research on UAV-aided data collection in IoT networks is carried out using a single UAV under the assumption that the UAV has sufficient energy. Some researchers believe that for large-scale linear scenarios, the scenario can be segmented according to the maximum available energy of the UAV, and in each sub-region, there is a ground station to control a single UAV for data collection using the proposed algorithms. However, the operation costs of multiple ground stations are relatively large, so it is practical to consider large-scale scenarios that only one ground station is set for multiple UAVs.

### B. Flying/Hovering Mode for UAV-aided Data Collection

Due to the advantages of mobility and controllability, UAVs are often used to aid data collection in IoT networks. They are equipped with communication modules that can communicate directly with ground sensors.

On the one hand, they can hover in the air to collect data, termed as *hovering mode*, where UAVs are designed to fly along the planned route and hover over the sensors in sequence to collect data [34]–[36]. In this mode, Basagni et

al [34] studied the performance of sensors under two UAV-aided data collection strategies, i.e., duty cycling and wake-up mechanism, with respect to the amount of data collected, the energy consumption of the sensors and their network lifetime. Ebrahimi et al. [35] studied the projection-based data collection problem, with the objective of minimizing the total transmission power and the length of the UAV's trajectory. Furthermore, UAVs don't have to hover directly above the sensors, they can stop at a suitable position to serve multiple sensors simultaneously [36].

On the other hand, UAVs can also collect data while flying, termed as *flying mode*. Zhan et al. [37] assumed that a UAV worked in the flying mode and discretized the UAV flight duration to multiple time slots. They jointly optimized the sensors' wake-up schedule and UAV's trajectory to minimize the maximum energy consumption of all sensors, while ensuring reliable and efficient data collection under the general channel fading model. Fan et al. [38] also used a UAV for data collection while flying, and they considered the data transmission power from the perspective of sensors, aiming to reduce the energy consumption of each sensor.

Generally speaking, the UAV-aided data collection in flying mode is more complicated than that in hovering mode, because the data transmission rate and time period of data collection will vary with the UAV's position and speed as in [39], [40]. Moreover, the selection of flying/hovering mode also depends on the features of the scenarios.

To sum up, within the scope of our knowledge, only a few of existing data collection solutions focus specially on the linear scenarios with multi-UAVs or multiple UAV flights. In this paper, we deploy ground cameras along the road and consider the data collection of these cameras. There may be many ground cameras, and thus a single UAV cannot complete all data collection work in one flight. For this reason, we study the UAV flight scheduling problem with cameras distributed along the road, aiming at minimizing the total number of UAV flights. Besides, in order to transmit video data efficiently, 60 GHz communication technology is adopted to provide higher transmission rate. Due to the short transmission range and weak signal penetration ability of 60 GHz communication link, the UAV needs to fly close to the cameras and communicates with them in flying and hovering mode.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present the system model and formulate the UAV flight scheduling problem. Under the constraints of UAV's energy and data storage, we aim to minimize the number of UAV flights to complete the data collection tasks of all the cameras. The main notations in this paper are summarized in Table I.

### A. System Model

A data collection system generally consists of a number of ground cameras (GCs), UAVs, a UAV ground station and a centralized controller. Let  $J = \{1, 2, \dots, n\}$  be the set of GCs and  $d_\alpha$  be the position of ground station. The centralized controller can periodically obtain the basic information of all

TABLE I  
SUMMARY OF NOTATIONS

Notation	Explanation
$n$	Number of ground cameras (GCs)
$j$	GC $j \in J$ , and $J = \{1, 2, \dots, n\}$
$i$	Flight candidate (FC) $i \in I$ , and $I = \{1, 2, \dots, n\}$
$d_\alpha$	Position of ground station
$d_j$	Distance between GC $j$ and GC 1
$v_{opt}$	Optimal UAV speed
$P_c$	Power consumption of data communication
$P_f$	Power consumption of flying
$P_h$	Power consumption of hovering
$E$	Total energy of UAV
$B$	Total data storage of UAV
$t_j$	Hovering time needed to serve GC $j$
$x_i$	Variable to indicate whether FC $i$ is activated
$y_{ij}$	Variable to indicate whether GC $j$ is assigned to FC $i$
$z_i^a$	Farthest position that FC $i$ can reach in the direction A
$z_i^b$	Farthest position that FC $i$ can reach in the direction B
$r_c$	Rate of data communication

cameras, including their positions and data size, then assign the UAVs to start from the ground station and complete data collection tasks. We consider that the GCs are placed in sequence along a road and the UAV is dispatched to collect video data from these cameras. The positions of GCs are denoted as  $d_1, d_2, \dots$ , and  $d_n$ , where  $d_1 = 0$  and  $d_j$  ( $j \geq 2$ ) represents the distance between GC  $j$  and GC 1. Let  $d_1 < d_2 < \dots < d_n$ . The ground station can be placed at any position along the linear infrastructure, i.e.,  $d_1 \leq d_\alpha \leq d_n$ .

The controller dispatches multiple UAV flights to collect data from GCs. In each flight, the UAV launches from the ground station and moves along the road with fixed altitude and speed. Let  $h$  be the flight altitude of the UAV. For a rotary-wing UAV with speed  $V$ , the propulsion power consumption can be approximated as a convex function [39], i.e.,

$$P(V) \approx P_0 \left(1 + \frac{3V^2}{U_{tip}^2}\right) + \frac{P_i V_0}{V} + \frac{1}{2} d_0 \rho s A V^3 \quad (1)$$

where  $P_0$  and  $P_i$  are two constants representing the blade profile power and induced power in hovering status,  $U_{tip}$  denotes the tip speed of the rotor blade,  $V_0$  is known as the mean rotor induced velocity in hover,  $d_0$  and  $s$  are the fuselage drag ratio and rotor solidity,  $\rho$  and  $A$  denote the air density and rotor disc area. In order to consume less energy to fly to the target GC, the UAV flies at the optimal speed  $v_{opt}$  that minimizes energy consumption per unit flight distance of UAV, i.e.,

$$v_{opt} = \arg \min_{V>0} \frac{P(V)}{V} \quad (2)$$

Since our goal is to minimize the number of UAV flights, the total energy of each UAV flight will be fully utilized. And the UAV needs to fly to the target camera and hover before each data collection. In such scenarios, communication related energy consumption only accounts for a small part of the total energy. What's more, the communication related energy

consumption is too small that it is not on the same order of magnitude with propulsion energy consumption [39]. So, we use a constant  $P_c$  to represent the power consumption of data communication. Let  $P_f$  and  $P_h$  denote the power consumption of flying and hovering, respectively. The number of GCs that can be served by a UAV flight is related to the total energy of UAV, denoted by  $E$ , and the total data storage of UAV, denoted by  $B$ .

As shown in Fig. 1, the UAV acts as an access point (AP) to provide 60 GHz network, and GCs are equipped with 802.11ad network adapters, acting as stations. The UAV can collect data in flying mode within the orange region in Fig. 1 and in hovering mode at a certain position close to the target GC. When the UAV flies near the target GC and the GC enters the UAV's communication range, the UAV will start beamforming to find the optimal communication link with the target GC [41], [42]. In detail, the UAV will conduct initiator transmit sector sweep and the GC performs in omnidirectional listening mode to determine the optimal transmitting sector from UAV to GC. The GC also determines the optimal transmitting sector from itself to UAV in the same manner. Then, the UAV will send request frames to the GC in omnidirectional mode, and the GC receives the frame from different directional beams to train its receiving sector. Next, the UAV trains its receiving sector in the same way. After that, the UAV and GC can determine their respective transmitting sector and receiving sector. Finally, the optimal communication link is determined by pairing them. When the UAV transmits data in the process of flying to the target GC, the transmission distance between them and the arrival angle of beam will constantly change with the movement of the UAV, resulting in link quality degradation. When UAV detects the link quality degradation, it will activate beam tracking and carries related requests in the header of the its frame [43]. Therefore, it can quickly test and switch to the beam with the best SNR while sending data.

Due to the limitation that 802.11ad technology can support a short transmission range only up to 10 meters and line-of-sight propagation [44], the communication range is not long enough so that it is normally impossible for the UAV to complete data collection of the data-intensive cameras when passing this range at the speed of  $v_{opt}$ . Therefore, data collection in the hovering mode is essential. Because very small amount of data is collected by the UAV in the flying mode, we only concentrate on the amount of data needed to be collected in the hovering mode and let  $t_j$  denote the hovering time needed to collect all data of GC  $j$ . For each GC, its data collection task cannot be divided, that is to say it should be completed by only one flight. Considering the energy limitation of UAV, we assume that each GC  $j$  satisfies  $2P_f d_j / v + (P_h + P_c) t_j \leq E$ . Similarly, considering the data storage limitation of UAV, we assume that each GC  $j$  satisfies  $t_j \cdot r_c \leq B$  where  $r_c$  denotes the rate of data communication.

## B. Problem Formulation

Due to the limitations of UAV's energy and data storage, a single UAV cannot serve a large number of GCs in a round flight, thus the data collection conducted by multiple flights

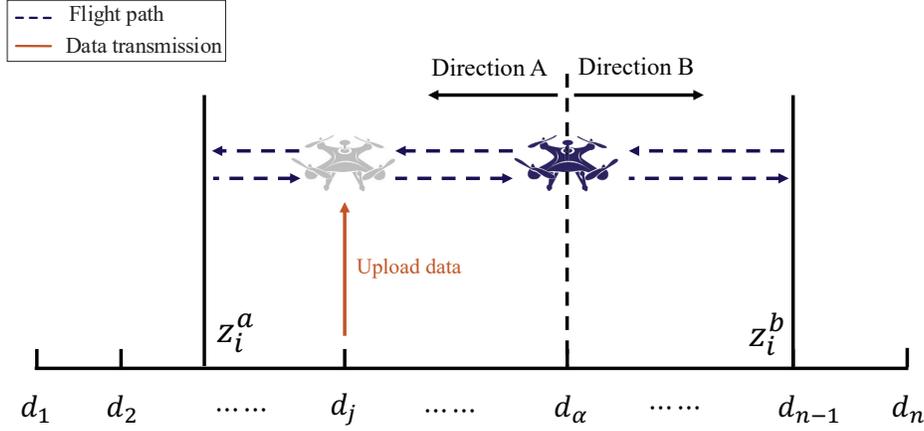


Fig. 2. An example for UAV-aided data collection during FC  $i$ , where the UAV flies between  $[z_i^a, z_i^b]$ . If GC  $j$  is assigned to this FC, the position of this GC must meet the restriction that  $z_i^a \leq d_j \leq z_i^b$ .

may be required. Moreover, data from one GC can only be collected during one flight, and it shouldn't be split to other flights. Therefore, at most  $n$  UAV flights are needed and let  $I = \{1, 2, \dots, n\}$  denote the set of UAV flight candidates (FCs) where part of FCs are activated as UAV flights to serve the GCs. It is necessary to design a UAV flight scheduling strategy so as to efficiently utilize the resources of UAV and minimize the number of UAV flights.

*Definition 1 (UFS Problem):* Given a UAV-aided data collection scenario with FC set  $I = \{1, 2, \dots, n\}$  and GC set  $J = \{1, 2, \dots, n\}$ , the UAV flight scheduling (UFS) problem is to determine the minimum number of the activated UAV flights, and find the proper FC-GC assignments while satisfying the data requirements of all GCs with the energy and data storage constraints of FCs.

A flight scheduling strategy would determine the minimum number of UAV flights and make FC-GC assignments for each GC to effectively reduce system costs, such as the costs produced by UAV operation and battery charging. Define  $x_i$  as a decision variable to indicate whether FC  $i$  is activated. That is,  $x_i = 1$  if FC  $i$  is activated and  $x_i = 0$  otherwise. Define  $y_{ij}$  as a decision variable to indicate if GC  $j$  is assigned to FC  $i$ . That is,  $y_{ij} = 1$  if the data collection task of GC  $j$  is assigned to FC  $i$  and  $y_{ij} = 0$  otherwise. Thus, we have

$$x_i = \{0, 1\}, \forall 1 \leq i \leq n. \quad (3)$$

$$y_{ij} = \{0, 1\}, \forall 1 \leq i \leq n, \forall 1 \leq j \leq n. \quad (4)$$

In the period, the data collected by GC  $j$  is required to be uploaded and this mission must be completed by only one flight. Here, each GC  $j$  is assigned to only one FC.

$$\sum_{i=1}^n y_{ij} = 1, \forall 1 \leq j \leq n. \quad (5)$$

Eq. (3), (4), and (5) are called the FC-GC assignment constraints.

If GC  $j$  is to be assigned to FC  $i$ , FC  $i$  must have been activated already. This is called the FC activation constraint.

$$y_{ij} \leq x_i, \forall 1 \leq i \leq n, \forall 1 \leq j \leq n. \quad (6)$$

A UAV launches from the ground station, and it has two flight directions along the line, i.e., the direction towards GC 1 (called direction A) and the direction towards GC  $n$  (called direction B), as shown in Fig. 2. In order to characterize the flight range of FC  $i$ , we define  $z_i^a$  as a decision variable to represent the farthest position FC  $i$  reaches in the direction A. Similarly, we define  $z_i^b$  as a decision variable to represent the farthest position FC  $i$  reaches in the direction B. We can see intuitively from Fig. 2 that

$$0 \leq z_i^a \leq d_\alpha \leq z_i^b \leq d_n, \forall 1 \leq i \leq n. \quad (7)$$

Let  $\delta_j$  denote whether GC  $j$  is located at the direction A of ground station. If GC  $j$  is located at the direction A of the ground station, we set  $\delta_j = 1$ , otherwise,  $\delta_j = 0$ , i.e.,

$$\delta_j = \begin{cases} 1 & d_j \leq d_\alpha \\ 0 & d_j > d_\alpha \end{cases} \quad (8)$$

If GC  $j$  is assigned to FC  $i$ , i.e.,  $y_{ij} = 1$ , a constraint should be satisfied firstly, that is, the GC is in the flying range of FC  $i$  that is determined by the values of  $z_i^a$  and  $z_i^b$ . Specifically, if GC  $j$  is located at the direction A of ground station,  $d_j$  should be in the range of  $[z_i^a, d_\alpha]$ . When GC  $j$  is located at the direction B,  $d_j$  should be in the range of  $[d_\alpha, z_i^b]$ . Thus, we have

$$\begin{aligned} & y_{ij} [(d_\alpha - d_j)\delta_j + (d_j - d_\alpha)(1 - \delta_j)] \\ & \leq (d_\alpha - z_i^a)\delta_j + (z_i^b - d_\alpha)(1 - \delta_j), \end{aligned} \quad (9)$$

$$\forall 1 \leq i \leq n, \forall 1 \leq j \leq n.$$

If GC  $j$  is not assigned to FC  $i$ , i.e.  $y_{ij} = 0$ , the constraint is always be satisfied. Eq. (7), (8), and (9) are called the FC range constraints.

Since the energy of a UAV is limited, the total consumption for UAV in FC  $i$  cannot exceed the total energy provided by the battery. The UAV energy constraint can be modeled as

$$\underbrace{(P_h + P_c) \sum_{j=1}^n t_j y_{ij}}_{\text{hovering and data communicating energy}} + \underbrace{2P_f \frac{z_i^b - z_i^a}{v_{opt}}}_{\text{flying energy}} \leq E, \forall 1 \leq i \leq n. \quad (10)$$

where  $P_h$  and  $P_f$  are two constants mentioned in the previous subsection representing the power consumption of hovering and flying, respectively.  $P_c$  denotes the power consumption of data communication.  $v_{opt}$  denotes the optimal speed that minimizes energy consumption per unit flight distance of UAV defined in Eq. (2). Other related parameters are explained in details in Table I. It is observed from Eq. (10) that the energy consumption consists of two components: one is the energy consumed for hovering and data communicating, the other is the energy consumed for flying.

Since the data storage of the UAV is also limited, the amount of collected data in FC  $i$  cannot exceed the total storage, i.e.,

$$r_c \sum_{j=1}^n t_j y_{ij} \leq B, \forall 1 \leq i \leq n. \quad (11)$$

where  $r_c$  denotes the rate of data communication and  $B$  denotes the total data storage.

Our objective is to minimize the number of FCs that are activated while all the GCs can be well served, i.e.,

$$\min \sum_{i=1}^n x_i \quad (12)$$

Therefore, the UFS problem can be formulated as an MILP model as follows.

$$\min \sum_{i=1}^n x_i$$

- s.t.* FC-GC assignment constraints: (3), (4), (5);  
 FC activation constraint: (6);  
 FC range constraints: (7), (8), (9);  
 UAV energy constraint: (10);  
 UAV data storage constraint: (11).

*Theorem 1:* The UFS problem is NP-hard.

*Proof:* The proof is conducted on a polynomial-time reduction from the classical bin packing problem (BPP) which is known to be NP-hard. Firstly, the UFS problem is an NP problem, because the solution can be verified in polynomial time. Secondly, we reduce the BPP as an example of the UFS problem. For a BPP, given a set  $\mathcal{I} = \{1, 2, \dots, n\}$  of  $n$  indivisible items, each of which has a specific weight  $w_i$  ( $i = 1, 2, \dots, n$ ), and  $M$  bins all having the same capacity  $c$ ,  $c \geq w_i$  for all  $i$ , the problem aims to select the minimum number of bins while each item can be packed into one of them [45]. We reduce the problem to the UFS problem as follows: each item in the BPP represents a GC, and the weight of item is defined by the time requirement of GC's data transmission, i.e.,  $w_i = t_i$ , ( $i = 1, 2, \dots, n$ ); each bin in the BPP represents a FC; the bin's capacity represents the corresponding FC's maximum service time, that is  $c = E/(P_h + P_c)$  if  $P_f = 0$  and  $B$  is adequate. Hence, we obtain a UFS instance ( $P_f = 0, B \rightarrow +\infty$ ) from the BPP. Since the BPP is NP-hard, we prove that the UFS problem is NP-hard.

However, the BPP is only a special case of the UFS problem, and the existing algorithms for the BPP cannot solve

our problem. Hence, it is necessary to come up with new algorithms.

#### IV. ALGORITHM DESIGN FOR SINGLE-DIRECTION UFS PROBLEM

Since the UFS problem is proved to be NP-hard, its optimal solution could not be obtained in polynomial time. This paper is devoted to finding an approximation algorithm for this problem and proving its approximation ratio. The concept of approximation ratio is usually defined as follows. For any input of size  $n$ , if the cost  $C$  of the solution by the algorithm is within a factor of  $\rho(n)$  of the cost  $C^*$  of an optimal solution, that is,  $\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$ , we can say that the algorithm has an approximation ratio of  $\rho(n)$  [46].

Firstly, we consider the UFS problem under the condition that the ground station is deployed at the position of GC 1, i.e.,  $d_\alpha = 0$ . Since the GCs are all in a single direction (direction B) of the ground station, the problem is called single-direction UFS (SUFS) problem.

In the following, we firstly relax the problem considering that the data collection task of each GC can be divided and arranged into multiple FCs, then propose the Iterated RSUFS algorithm to solve the problem. A feasible solution to the SUFS problem can be obtained by adjusting the results obtained by RSUFS. On this basis, we design the NF\_SUFS algorithm and prove its constant approximation. Finally, we propose an improved algorithm called FF\_SUFS, which makes better use of the remaining available time of UAV. To be intuitive, we illustrate the execution results of these four algorithms through an example in Fig. 3.

##### A. Algorithm RSUFS\_to\_SUFS

According to the definition of the SUFS problem, the data collection task of each GC cannot be divided and would be assigned to one and only one flight. We relax the constraint and allow GC's data collection task can be divided and assigned to multiple flights. The relaxed problem is called the RSUFS problem.

As described above, at most  $n$  flights are needed, and FCs  $\{1, 2, \dots, n\}$  are provided for selection. To save energy and put more energy into data collection, the UAV only needs to fly to the farthest GC that is assigned to this flight, and turns back. So, the farthest served GC of any two FCs would be distinct. On this basis, we can think that the farthest served GC of FC  $i$  is GC  $i$ . In other words, if FC  $i$  is activated, GC  $i$  must be served by FC  $i$ . When the maximum flight distance is determined, the energy available for hovering to collect data will also be determined. Let  $T_i$  denote the maximum time available for data communication which depends on the remaining energy and storage capacity of the FC  $i$ , i.e.,

$$T_i = \min\left( \underbrace{\frac{E - \frac{2P_f d_i}{v_{opt}}}{P_h + P_c}}_{\text{depending on energy}}, \underbrace{\frac{B}{r_c}}_{\text{depending on storage}} \right). \quad (13)$$

Since the flight distance increases as the index  $i$  increases, we can get that  $T_1 \geq T_2 \geq \dots \geq T_n$ . In the RSUFS problem,

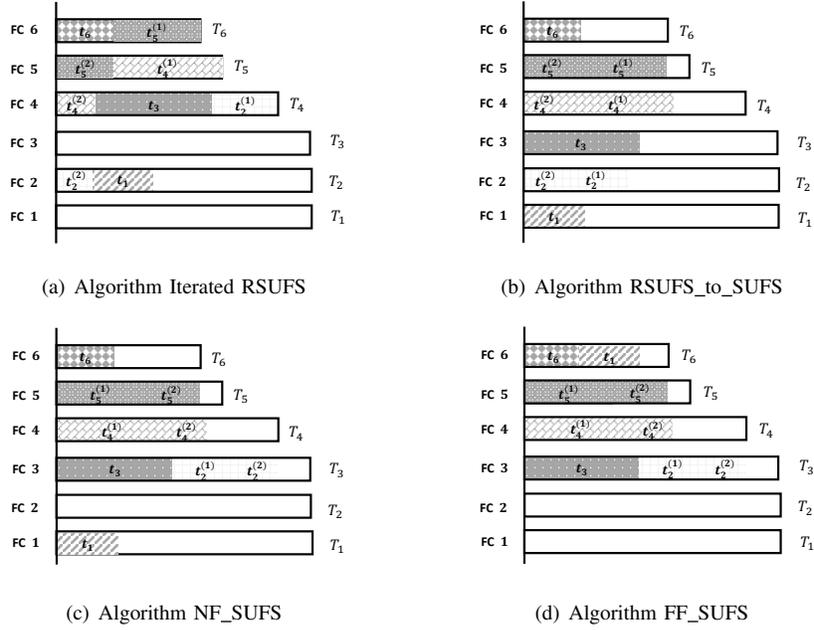


Fig. 3. An illustration of the execution results of the four algorithms. In this example, there are six GCs in total, so at most six FCs are needed. According to the definition of FC, we use six rectangles of different length to represent the time that FC can utilize for hovering to do data transmission, and let  $T_i$  denote the available data communication time of FC  $i$ . For  $i \in [4, 6]$ , the time  $T_i$  is varying, because UAV flying consumes more energy to serve long-distance GCs so that the remaining energy is not enough relative to storage capacity, and  $T_i$  is limited by varying remaining energy. For  $i \in [1, 3]$ , the available time  $T_i$  is the same, because UAV flying consumes less energy, and remaining energy for hovering and data communicating is abundant so that  $T_i$  is limited by fixed storage capacity. Because the Iterated RSUFS algorithm divides the data collection tasks of GCs, some GCs' service time will be divided into two parts such as  $t_5^{(1)}$  and  $t_5^{(2)}$  in Fig. 3(a). By comparing the number of activated FCs in Fig. 3(b), Fig. 3(c), and Fig. 3(d), it can be found that FF\_SUFS algorithm obtains the best result with only 4 activated FCs.

a GC can be assigned to multiple flights, and each flight completes part of its data collection task.

We design an iterative algorithm for the RSUFS problem, called Iterated RSUFS. Let tuple  $(j, t_{ij})$  represent the FC-GC assignment where FC  $i$  takes time  $t_{ij}$  to serve GC  $j$ . We find that FC  $n$  must be activated so that GC  $n$  can be served. In each iteration, we determine the farthest uncompleted GC  $j$ , and activate FC  $j$  so that GC  $j$ 's data collection task can be completed, because FC  $j$  is the one with the most residual energy which can serve GC  $j$ . Meanwhile, other GCs in descending order of  $|d_j - d_\alpha|$  are assigned as many as possible. In addition, the last assigned GC of each iteration maybe partially assigned to the current FC, and would be completed in the next iteration.

*Lemma 1:* The Iterated RSUFS algorithm can obtain an optimal solution for the RSUFS problem.

*Proof:* Let  $U_{IR}$  be the set of activated UAV flights determined by the Iterated RSUFS algorithm. For any solution  $U'$  of the RSUFS problem, we want to prove that  $|U'| \geq |U_{IR}|$ . Let  $U_{IR} = \{i_1, i_2, \dots, i_p\} \subset I$ , where  $i_1 > i_2 > \dots > i_p$ . Let  $U' = \{i'_1, i'_2, \dots, i'_q\} \subset I$ , where  $i'_1 > i'_2 > \dots > i'_q$ .

According to the Iterated RSUFS algorithm, we start from GC  $n$  and analyze in the order of decreasing distance. When there exists  $k$  that makes  $i_k \neq i'_k$  and  $i_s = i'_s$  for all  $s < k$ , we can get that  $i_k \leq i'_k$ , otherwise the data collection task of GC  $i_k$  cannot be completed in the solution  $U'$ . Because the Iterated RSUFS algorithm makes full use of UAVs' service time, we can get that  $T_{i_1} + T_{i_2} + \dots + T_{i_{k-1}} < \sum_{j=i_k}^n t_j$ . If  $i_k > i'_k$ , GC  $i_k$  must be fully served before UAV flight  $i'_k$ , which is

---

#### Algorithm 1: Iterated RSUFS

---

**input :** GCs  $J = \{1, 2, \dots, n\}$  with  $t_j$ ;  
 $\{T_1, T_2, \dots, T_n\}$   
**output:** UAV flights  $\mathcal{U}$  and FC-GC assignments  $\mathcal{R}$

```

1 begin
2    $\mathcal{U} = \{n\}, \mathcal{R} = \emptyset;$ 
3    $i = n, T' = T_i, \mathcal{R}_i = \emptyset;$ 
4   for  $j = n$  to 1 do
5     if  $t_j \leq T'$  then
6        $\mathcal{R}_i = \mathcal{R}_i \cup \{(j, t_j)\};$ 
7        $T' = T' - t_j;$ 
8     else
9       if  $T' \neq 0$  then
10         $\mathcal{R}_i = \mathcal{R}_i \cup \{(j, T')\};$ 
11         $t_j = t_j - T';$ 
12      end
13       $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_i\};$ 
14       $i = j, T' = T_i - t_j, \mathcal{U} = \mathcal{U} \cup \{i\};$ 
15       $\mathcal{R}_i = \mathcal{R}_i \cup \{(j, t_j)\};$ 
16    end
17  end
18   $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_i\};$ 
19  return  $\mathcal{U}$  and  $\mathcal{R};$ 
20 end

```

---

infeasible. Similarly,  $i_s \leq i'_s$  for all  $s > k \wedge s \leq \min\{p, q\}$ .  $T_{i_1} + T_{i_2} + \dots + T_{i_{p-1}} + T_p \geq \sum_{j=1}^n t_j$  and  $T_{i_1} + T_{i_2} + \dots + T_{i_{p-1}} < \sum_{j=1}^n t_j$ . If  $q < p$ ,  $T_{i'_1} + T_{i'_2} + \dots + T_{i'_q} \leq T_{i_1} + T_{i_2} + \dots + T_{i_q} \leq T_{i_1} + T_{i_2} + \dots + T_{i_{p-1}} < \sum_{j=1}^n t_j$ , so that it is impossible. Hence, we get that  $q \geq p$ .

When there does not exist  $k$  that makes  $i_k \neq i'_k$  and  $i_s = i'_s$  for all  $s < k$ , that is  $i_s = i'_s$  for all  $s \leq \min\{p, q\}$ . Thus, we can get that  $q \geq p$ . That is because  $T_{i'_1} + T_{i'_2} + \dots + T_{i'_q} \leq \sum_{j=1}^n t_j$  and the solution  $U'$  is not feasible if  $q < p$ .

As discussed above, we can obtain that  $q \geq p$ , and the solution of the Iterated RSUFS algorithm is optimal.

Now, we propose the RSUFS\_to\_SUFS algorithm that can transfer the solution of the RSUFS problem to that of the SUFS problem. The set of initial UAV flights and set of initial FC-GC assignments are respectively set to be  $\mathcal{U}_{IR}$  and  $\mathcal{R}_{IR}$ , which are obtained from the Iterated RSUFS algorithm. For each UAV flight  $i_k$ , all tuples in  $\mathcal{R}_{i_k}$  except for GC  $i_k$ 's tuple are moved into  $\mathcal{R}_{i_k-1}$ , and if the UAV flight  $i_k - 1$  has not been activated, it will be activated at the same time. In addition, if there exists a tuple of GC  $i_k$  in  $\mathcal{R}_{i_k-1}$ , it will be merged with the corresponding tuple in  $\mathcal{R}_{i_k}$ . See Algorithm 2 for more details.

---

**Algorithm 2: RSUFS\_to\_SUFS**


---

**input :** GCs  $J = \{1, 2, \dots, n\}$  with  $t_j$ ;  
 $\{T_1, T_2, \dots, T_n\}$ ; The solution of Iterated  
 RSUFS  $\mathcal{U}_{IR} = \{i_1, i_2, \dots, i_p\}$  and  $\mathcal{R}_{IR}$

**output:** UAV flights  $\mathcal{U}$  and FC-GC assignments  $\mathcal{R}$

```

1 begin
2    $\mathcal{U} = \mathcal{U}_{IR}, \mathcal{R} = \mathcal{R}_{IR}, k = p, j_o = 1$ ;
3   while  $k \geq 2$  do
4     if  $\exists f, j_o \leq f < i_k$  then
5        $\mathcal{U} = \mathcal{U} \cup \{i_k - 1\}$ ;
6       move all  $(f, t_f)$  from  $\mathcal{R}_{i_k}$  to  $\mathcal{R}_{i_k-1}$ ;
7        $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_{i_k-1}\}$ ;
8     end
9     if  $\exists (i_k, t_{i_k}^{(1)}), (i_k, t_{i_k}^{(1)}) \in \mathcal{R}_{i_k-1}$  then
10      merge  $(i_k, t_{i_k}^{(1)})$  into  $\mathcal{R}_{i_k}$ ;
11    end
12     $j_o = i_k + 1, k = k - 1$ ;
13  end
14  return  $\mathcal{U}$  and  $\mathcal{R}$ ;
15 end
```

---

*Lemma 2:* The number of activated UAV flights obtained by the RSUFS\_to\_SUFS algorithm would not exceed two times of the RSUFS algorithm, that is  $|\mathcal{U}_{RTS}| < 2|\mathcal{U}_{IR}|$ .

*Proof:* Let  $p = |\mathcal{U}_{IR}|$ . As shown in the RSUFS\_to\_SUFS algorithm, the number of iterations is  $p-1$ . In each iteration, at most one UAV flight is activated. Thus,  $|\mathcal{U}_{RTS}| \leq p + p - 1 < 2p$ , i.e.,  $|\mathcal{U}_{RTS}| < 2|\mathcal{U}_{IR}|$ .

### B. Algorithm NF\_SUFS

Through analysis, we find that the RSUFS\_to\_SUFS algorithm does not make full use of the remaining available time of activated UAV flights in the process of FC-GC assignments,

resulting in the increment of the number of UAV flights. In order to reduce the number of activated flights as much as possible, we design an algorithm called NF\_SUFS, which iteratively activates UAV flights and each activated flight should be fully utilized by assigning as many as GCs to it. Since the UAV's available service time is determined by its data storage and the distance to the farthest GC, we find that the FCs can be abstracted as bins with different capacities, and the next-fit strategy of bin packing is adopted in the NF\_SUFS algorithm. The algorithm iteratively assigns the GCs in the order of decreasing its distance. In each iteration, a GC is assigned to the current activated flight, and a new flight is activated if the current flight is not available. More detailed, we start from the farthest GC and assume that GC  $j$  can be assigned to flight  $i$ , for the next GC  $j-1$ , if its service time of  $t_{j-1}$  is smaller than the remaining available time of flight  $i$ , then it can be assigned to flight  $i$ , otherwise, it will activate a new flight whose farthest serving point is GC  $j-1$ . Repeat until all GCs are assigned.

---

**Algorithm 3: NF\_SUFS**


---

**input :** GCs  $J = \{1, 2, \dots, n\}$  with  $t_j$ ;  
 $\{T_1, T_2, \dots, T_n\}$

**output:** UAV flights  $\mathcal{U}$  and FC-GC assignments  $\mathcal{R}$

```

1 begin
2    $\mathcal{U} = \{n\}, \mathcal{R} = \emptyset$ ;
3    $i = n, T' = T_j, \mathcal{R}_i = \emptyset$ ;
4   for  $j = n$  to 1 do
5      $T' = T' - t_j, \mathcal{R}_i = \mathcal{R}_i \cup \{j\}$ ;
6     if  $j > 1$  then
7       if  $T' < t_{j-1}$  then
8          $\mathcal{R} = \mathcal{R} \cup \mathcal{R}_i$ ;
9          $i = j - 1, T' = T_i, \mathcal{R}_i = \emptyset$ ,
10         $\mathcal{U} = \mathcal{U} \cup \{i\}$ ;
11      end
12    else
13       $\mathcal{R} = \mathcal{R} \cup \mathcal{R}_i$ ;
14    end
15  end
16  return  $\mathcal{U}$  and  $\mathcal{R}$ ;
17 end
```

---

Next, we analyze the performance of the Algorithm NF\_SUFS on the basis of Lemma 1 and Lemma 2.

*Theorem 2:* The approximation ratio of NF\_SUFS algorithm is 2.

*Proof:* Let  $\mathcal{U}_{NS}$  be the solution of the NF\_SUFS Algorithm. Let  $OPT$  be the number of UAV flights determined by the optimal solution of the SUFS problem. Let  $OPT^r$  be the number of activated flights determined by the optimal solution of the RSUFS problem. We can obtain that  $OPT^r \leq OPT$ . Let  $\mathcal{U}_{IR}$  be the solution of Iterated RSUFS algorithm. Let  $\mathcal{U}_{RTS}$  be the solution of RSUFS\_to\_SUFS algorithm. According to Lemma 1 and 2, we get that  $|\mathcal{U}_{RTS}| < 2|\mathcal{U}_{IR}| = 2OPT^r$ . Since we can easily find that  $|\mathcal{U}_{NS}| \leq |\mathcal{U}_{RTS}|$ , we obtain that  $|\mathcal{U}_{NS}| \leq |\mathcal{U}_{RTS}| < 2|\mathcal{U}_{IR}| = 2OPT^r \leq 2OPT$ . Thus,  $|\mathcal{U}_{NS}| \leq 2OPT$ .

### C. Algorithm FF\_SUFS

In order to further improve algorithm performance, we try to adopt the first-fit strategy of bin packing, and propose the corresponding algorithm called FF\_SUFS. The FF\_SUFS algorithm iteratively assigns the GCs in the order of decreasing their distance. In each iteration, it assigns the GC to the first appropriate activated flight and activates one new flight if no activated flight is available. Repeat until all GCs are assigned. Compared with the NF\_SUFS algorithm, for the assignment of an GC  $j$ , the FF\_SUFS algorithm will check all the activated flights in the activation order, while the NF\_SUFS algorithm only checks the last activated flight.

Since the FF\_SUFS algorithm utilizes the remaining available time of UAVs more reasonably than the NF\_SUFS algorithm, the approximation ratio of FF\_SUFS is no more than 2 which is obtained in Theorem 2.

---

#### Algorithm 4: FF\_SUFS

---

**input :** GCs  $J = \{1, 2, \dots, n\}$  with  $t_j$ ;  $\{T_1, T_2, \dots, T_n\}$   
**output:** UAV flights  $\mathcal{U}$  and FC-GC assignments  $\mathcal{R}$

```

1 begin
2    $\mathcal{U} = \emptyset, \mathcal{R} = \emptyset$ ;
3   for  $j = n$  to 1 do
4      $id = j$ ;
5     foreach  $i \in \mathcal{U}$  do
6       if  $T'_i \geq t_j$  then
7          $id = i$ ;
8         break;
9     end
10    end
11    if  $id == j$  then
12       $\mathcal{U} = \mathcal{U} \cup \{j\}, \mathcal{R}_j = \emptyset, T'_j = T_j$ ;
13    end
14     $T'_{id} = T'_{id} - t_j, \mathcal{R}_{id} = \mathcal{R}_{id} \cup \{j\}$ ;
15  end
16  foreach  $i \in \mathcal{U}$  do
17     $\mathcal{R} = \mathcal{R} \cup \mathcal{R}_i$ 
18  end
19  return  $\mathcal{U}$  and  $\mathcal{R}$ ;
20 end

```

---

## V. ALGORITHM DESIGN FOR GENERAL UFS PROBLEM

In this section, we consider the general UFS problem where the ground station is deployed at any position between GC 1 and GC  $n$ , i.e.,  $d_1 \leq d_\alpha \leq d_n$ , and propose a constant approximation algorithm for this problem.

### A. Algorithm FF\_UFS

We propose an algorithm called FF\_UFS, and its basic idea is to assign UAV flights for data collection in two directions, respectively. As shown in Algorithm 5, we first find  $k$  so that  $d_k \leq d_\alpha < d_{k+1}$ , and divide the GCs into two groups, i.e.,  $\{1, 2, \dots, k\}$  and  $\{k+1, k+2, \dots, n\}$ . We first invoke the FF\_SUFS algorithm to deal with GCs in  $\{1, 2, \dots, k\}$  (the

corresponding set  $J_a$ ), and obtain the set of UAV flights  $\mathcal{U}^a$ . In order to minimize the number of UAV flights, we then use the remaining available time of flights in  $\mathcal{U}^a$  to serve the GCs in  $\{k+1, k+2, \dots, n\}$  (the corresponding set  $J_b$ ). More detailed, we sort the flights according to the residual energy and available time in the descending order and start from the farthest GC and consider whether the UAV can fly to serve it and return back to the ground station, and so on. Finally, we deal with the rest GCs in  $\{k+1, k+2, \dots, n\}$  (the corresponding set  $J'_b$ ) by invoking the FF\_SUFS algorithm again, and obtain the set of UAV flights  $\mathcal{U}^b$ . Thus, we can get the final set of UAV flights  $\mathcal{U}$ , i.e.,  $\mathcal{U}^a \cup \mathcal{U}^b$ .

### B. Approximation of FF\_UFS

*Theorem 3:* The approximation ratio of FF\_UFS algorithm is 3.

*Proof:* Let  $\mathcal{U}_g$  be the solution of FF\_UFS algorithm, and  $ALG_g = |\mathcal{U}_g|$ . Let  $\mathcal{U}_g^{opt}$  be the optimal solution of the general UFS problem, and  $OPT_g = |\mathcal{U}_g^{opt}|$ . Let  $\mathcal{U}_s(J_a)$  be the solution of FF\_SUFS algorithm with  $J_a$ , and  $ALG_s(J_a) = |\mathcal{U}_s(J_a)|$ . Let  $\mathcal{U}_s(J_a)^{opt}$  be the optimal value of the SUFS problem with  $J_a$ , and  $OPT_s(J_a) = |\mathcal{U}_s(J_a)^{opt}|$ . Hence, we get:

$$\begin{aligned}
 ALG_g &= ALG_s(J_a) + ALG_s(J'_b) \\
 &\leq ALG_s(J_a) + ALG_s(J_b) \\
 &\leq 2OPT_s(J_a) + 2OPT_s(J_b)
 \end{aligned} \tag{14}$$

For the optimal solution  $\mathcal{U}_g^{opt}$ , there are some flights that the UAV only flies to serve GCs in direction A, i.e.,  $\mathcal{U}_{ga}^{opt}$ . There are some flights that the UAV only flies to serve in direction B, i.e.,  $\mathcal{U}_{gb}^{opt}$ . There are also some flights that the UAV flies across the two directions, i.e.,  $\mathcal{U}_{gc}^{opt}$ . Let  $\mathcal{U}_{ga}^{opt'} = \mathcal{U}_{ga}^{opt}$  and  $\mathcal{U}_{gb}^{opt'} = \mathcal{U}_{gb}^{opt}$ . Let  $\{i_{c_1}, i_{c_2}, \dots, i_{c_k}\}$  be the set of  $\mathcal{U}_{gc}^{opt}$ . For each flight  $i_{c_p}$ , the energy is consumed in two directions, i.e.,  $E_{c_p}^a$  and  $E_{c_p}^b$ . If  $E_{c_p}^a \geq E_{c_p}^b$ , flight  $i_{c_p}$  will not serve the GCs in direction B and  $\mathcal{U}_{ga}^{opt'} = \mathcal{U}_{ga}^{opt'} \cup \{i_{c_p}\}$ . Otherwise, flight  $i_{c_p}$  will not serve the GCs in direction A and  $\mathcal{U}_{gb}^{opt'} = \mathcal{U}_{gb}^{opt'} \cup \{i_{c_p}\}$ . Let  $k_a$  be the number of flights added into  $\mathcal{U}_{ga}^{opt'}$ . Thus, there would be  $k - k_a$  groups of GCs in direction A and  $k_a$  groups of GCs in direction B are not served. We need to arrange at most  $\lceil \frac{k - k_a}{2} \rceil$  flights to serve these GCs in direction A and  $\lceil \frac{k_a}{2} \rceil$  to serve the GCs in direction B. As discussed above, we construct a solution where no UAV flight across the two directions, and the number of flights is  $|\mathcal{U}_{ga}^{opt'}| + |\mathcal{U}_{gb}^{opt'}| + k'$ , where  $k' \leq \lceil \frac{k - k_a}{2} \rceil + \lceil \frac{k_a}{2} \rceil$  and  $|\mathcal{U}_g^{opt}| = |\mathcal{U}_{ga}^{opt'}| + |\mathcal{U}_{gb}^{opt'}|$ . In additional, we can get:

$$\begin{aligned}
 |\mathcal{U}_{ga}^{opt'}| + |\mathcal{U}_{gb}^{opt'}| + k' &\leq |\mathcal{U}_g^{opt}| + \lceil \frac{k - k_a}{2} \rceil + \lceil \frac{k_a}{2} \rceil \\
 &\leq |\mathcal{U}_g^{opt}| + \frac{k}{2} + 1 \leq 1.5|\mathcal{U}_g^{opt}| + 1 = 1.5OPT_g + 1
 \end{aligned} \tag{15}$$

Since  $OPT_s(J_a) + OPT_s(J_b) \leq |\mathcal{U}_{ga}^{opt'}| + |\mathcal{U}_{gb}^{opt'}| + k'$ , we merge Eq. (14) and Eq. (15) as follows:

$$\begin{aligned}
 ALG_g &\leq 2(OPT_s(J_a) + OPT_s(J_b)) \leq 2(|\mathcal{U}_{ga}^{opt'}| \\
 &\quad + |\mathcal{U}_{gb}^{opt'}| + k') \leq 3OPT_g + 2
 \end{aligned} \tag{16}$$

**Algorithm 5: FF\_UFS**


---

```

input : GCs  $J = \{1, 2, \dots, n\}$  with  $d_j$  and  $t_j$ ;
         Ground station  $d_\alpha$ ; UAV parameter  $E, v_{opt},$ 
          $P_c, P_f,$  and  $P_h$ 
output: UAV flights  $\mathcal{U}$  and FC-GC assignments  $\mathcal{R}$ 
1 begin
2    $\mathcal{U} = \emptyset, \mathcal{R} = \emptyset, m = 0;$ 
3   find  $k$  so that  $d_k \leq d_\alpha < d_{k+1}$ ;
4   /* deal with GCs  $\{1, 2, \dots, k\}$  */
5    $J_a = \{1, 2, \dots, k\}$ ;
6   invoke Algorithm FF-SUFS with  $J_a,$  and obtain
    $\mathcal{U}^a, \mathcal{R}^a,$  remaining time  $\{T_1^a, T_2^a, \dots, T_m^a\}$  and
   remaining energy  $\{E_1^a, E_2^a, \dots, E_m^a\}$ ;
7   /* deal with GCs  $\{k+1, k+2, \dots, n\}$  */
8    $J_b = \{k+1, k+2, \dots, n\}$ ;
9   sort  $\mathcal{U}^a$  by decreasing order according to  $E_i^a$  and
    $T_i^a,$  let  $\{i_1, i_2, \dots, i_m\}$  denote the sorted result;
10   $f = 1, j = n;$ 
11  while  $f \leq m$  and  $j \geq k$  do
12    if  $j \in J_b$  and
        $E_{i_f}^a \geq \frac{2P_f(d_j - d_\alpha)}{v_{opt}} + (P_h + P_c)t_j$  and
        $T_{i_f}^a \geq t_j$  then
13      FC  $i_f$  can fly farthest to GC  $j$ ;
14      assign GC  $j$  to  $\mathcal{R}_{i_f},$  update  $T_{i_f}^a, E_{i_f}^a;$ 
15       $J_b = J_b \setminus \{j\};$ 
16       $T_{i_f}^a = \min(T_{i_f}^a, \frac{E_{i_f}^a}{P_h + P_c});$ 
17      for  $s = j - 1$  to 1 do
18        if  $s \in J_b$  and  $T_{i_f}^a \geq t_s$  then
19           $J_b = J_b \setminus \{s\};$ 
20          assign GC  $j$  to  $\mathcal{R}_{i_f},$  update  $T_{i_f}^a,$ 
            $E_{i_f}^a;$ 
21        end
22      end
23       $f = f + 1;$ 
24    end
25    else
26       $j = j - 1;$ 
27    end
28  end
29  let  $J'_b$  denote the rest of  $J_b,$  invoke Algorithm
   FF-SUFS with  $J'_b,$  and obtain  $\mathcal{U}^b$  and  $\mathcal{R}^b$ ;
30   $\mathcal{U} = \mathcal{U}^a \cup \mathcal{U}^b, \mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{|\mathcal{U}|}\};$ 
31  return  $\mathcal{U}$  and  $\mathcal{R}$ ;
32 end

```

---

## VI. PERFORMANCE EVALUATION

The theoretical analysis has verified the worst-case performance bounds of the proposed algorithms. In this section, we conduct simulations to further evaluate the average performance of our proposed algorithms. As studied above in Eq. (13), for each UAV flight, the available data communication time is limited by UAV's energy and storage capacity. In real scenarios, the storage capacity can be a fixed constant for each FC, but the energy used for hovering and data communicating

TABLE II  
EXPERIMENTAL PARAMETERS AND VALUES

Notation	Explanation	Value
$E$	Total UAV energy ( $J$ )	450000
$P_h$	Power consumption of hovering ( $W$ )	168
$P_c$	Power consumption of data communication ( $W$ )	5
$v$	Optimal UAV speed ( $m/s$ )	18
$P_v$	Power consumption of flying at speed $v$ ( $W$ )	159

varies with the UAV flying distance. Hence, we consider that the storage capacity is adequate and only concentrate on the constraint of UAV energy in our simulations.

## A. Simulation Settings

We consider road scenarios for UAV-aided data collection. We set the length of the road to be 10 km, and there are from 10 to 500 GCs evenly deployed along the road. The distance between adjacent GCs is roughly the same, allowing a fluctuation of less than 10 meters. Referring to the UAV's parameter settings [39], the notations and their corresponding values used in our simulations are summarized in Table II. For each GC, the amount of data that needs to be collected may be different, due to individual function requirements. Hence, we set a random service time for each GC, which is subject to the random distribution in (1, 10) minutes.

To evaluate the performance of the proposed algorithms, we present the MILP-based solution as a benchmark, which can obtain the near-optimal solution of UFS problem by using Gurobi [47] to solve the MILP model directly. In addition, we also compare with another benchmark algorithm *approAlgNoNei* [32] that provides an heuristic solution by finding all TSP paths within the edge weight threshold and segmenting them with the limitation of the UAV's total energy.

For each setting, we randomly generate 20 different scenarios to run our algorithms and the compared algorithms, and take the average as the final results.

## B. Evaluation of FF\_SUFS and NF\_SUFS algorithms

In this part of simulations, we consider the scenarios where the ground station is deployed at the left end of the road. We run the algorithms FF\_SUFS, NF\_SUFS, MILP-based, and *approAlgNoNei*, and evaluate the performance in terms of the number of flights and computation time. The results are shown in Table III. It can be seen from the table that there is only a small gap between our solutions and MILP-based solutions on the performance of number of flights, which validates the near optimality of our algorithms. When the number of GCs becomes large (e.g.,  $n = 120$ ), the MILP model cannot be solved within an acceptable time, but our proposed algorithms can still provide solutions efficiently. In addition, we find that algorithm FF\_SUFS performs better than NF\_SUFS and *approAlgNoNei*, and this advantage becomes more significant as the number of GCs  $n$  increases.

We also compare the results gotten from FF\_SUFS, NF\_SUFS and *approAlgNoNei* when increasing the number

TABLE III  
PERFORMANCE OF FF\_SUFS, NF\_SUFS, MILP-BASED AND APPROALGNO NEI FOR THE SUFS PROBLEM

Number of GCs	Algorithm		FF_SUFS		NF_SUFS		MILP-based		approAlgNoNei	
	Flights	Time( <i>ms</i> )	Flights	Time( <i>ms</i> )	Flights	Time( <i>ms</i> )	Flights	Time( <i>ms</i> )	Flights	Time( <i>ms</i> )
$n = 10$	2.55	0.02	2.60	0.01	2.55	36.00	2.80	1.79		
$n = 40$	7.45	0.07	8.15	0.06	7.45	$2.86 \times 10^2$	9.00	32.42		
$n = 80$	14.20	0.13	15.55	0.11	14.15	$2.39 \times 10^3$	17.25	$1.69 \times 10^2$		
$n = 120$	21.45	0.20	23.55	0.17	-	-	25.70	$4.67 \times 10^2$		

of GCs  $n$  to a large value. As shown in Fig. 4, the number of flights obtained by these three algorithms shows an almost linear growth trend with the increment of  $n$ . Moreover, the two curves in this figure show that the number of flights obtained by FF\_SUFS are about 16% to 17.6% less than approAlgNoNei, and the number of flights obtained by NF\_SUFS are about 7.8% to 10.6% less than approAlgNoNei. Hence, FF\_SUFS shows better performance, and the larger  $n$  is, the bigger the gap between FF\_SUFS and the other two algorithms is. That is because when assigning a GC to flights, FF\_SUFS will consider all the flights that have been activated and make full use of their remaining energy.

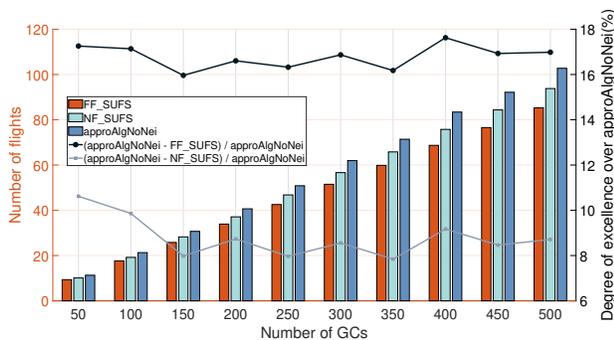


Fig. 4. Performance of FF\_SUFS, NF\_SUFS and approAlgNoNei varying the number of GCs.

All the experimental results agree with the constant approximation ratio we have proved. It can be seen more clearly from Table III that the results of FF\_SUFS and NF\_SUFS are close to the results of MILP-based solution, much less than our proved approximation ratio 2.

### C. Evaluation of FF\_UFS algorithm

In this part of simulations, we deploy the ground station at the middle of the road, i.e.,  $d_\alpha = 5000m$ , and then show the performance of FF\_UFS, MILP-based solution and approAlgNoNei in Table IV. It can be found that when the ground station is deployed at the middle of the path, the problem becomes more complicated than the case when the ground station is deploy at one end of the road, and the computation time of these three algorithms increases significantly. When the number of GCs  $n$  reaches 80, the MILP-based solution can no longer obtain the results within an acceptable time. Similar with Table III, the results gotten from FF\_UFS are very close to the near-optimal solution gotten from MILP-based solution and better than the results gotten

from approAlgNoNei. Moreover, when the number of GCs  $n$  becomes larger, the advantage of FF\_UFS algorithm becomes more significant.

We also compare the performance of FF\_UFS and approAlgNoNei varying the number of GCs. As shown in Fig. 5, the number of UAV flights required by FF\_UFS and approAlgNoNei shows a trend of almost linear growth. Meanwhile, FF\_UFS performs better than approAlgNoNei, and the gap between them becomes larger along with the growth of  $n$ . When  $n = 500$ , the gap reaches 13.6.

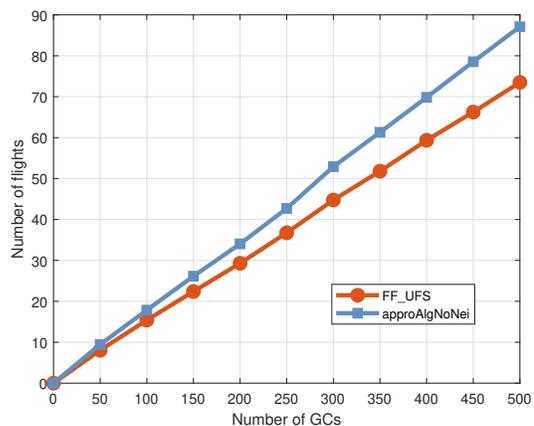


Fig. 5. Performance of FF\_UFS and approAlgNoNei varying the number of GCs .

To observe the performance of algorithms when varying the position of ground station, we set 11 different positions for ground station, i.e.,  $d_\alpha = 1000 * i$ , where  $i \in [0, 10]$ . As we can see intuitively from Fig. 6, the number of flights is smallest when  $d_\alpha = 5000$ , and grows larger when the ground station moves from the middle to the both ends. Moreover, Fig. 6(a)

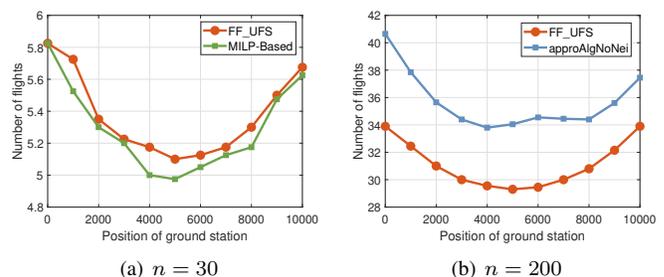


Fig. 6. Performance of FF\_UFS, MILP-based and approAlgNoNei varying position of ground station.

TABLE IV  
PERFORMANCE OF FF\_UFS, MILP-BASED AND APPROALGNOEI WHEN GROUND STATION IS AT THE MIDDLE POSITION

Number of GCs	Algorithm		FF_UFS		MILP-based		approAlgNoNei	
	Flights	Time(ms)	Flights	Time(ms)	Flights	Time(ms)	Flights	Time(ms)
$n = 10$	2.20	0.04	2.10	51.9	2.35	1.70		
$n = 40$	6.50	0.12	6.45	$4.10 \times 10^4$	7.45	33.69		
$n = 80$	12.40	0.21	-	-	14.50	$1.75 \times 10^2$		
$n = 120$	18.50	0.34	-	-	21.60	$4.78 \times 10^2$		

confirms that no matter where the ground station is located, the correctness of the approximation ratio proved above will not be affected. In the simulation scenarios of Table IV and Fig. 6(a), the average performance of FF\_UFS algorithm is at most 4.8% higher than the near optimal solutions gotten from MILP model, which is better than the theoretical approximation ratio.

In Fig. 7, we fix  $n = 200$  and  $d_\alpha = 5000m$ , and vary the mean values of GCs' service time to find its impact on the number of UAV flights required. We denoted the mean value of GCs' service time by  $m$  and set  $m \in [2, 12]$ . Given the value  $m$ , the service time of each GC is set to be uniformly distributed in  $[1, 2 * m]$ . Apparently, with the increase of mean value of GCs' service time, both the FF\_UFS and approAlgNoNei algorithms need more flights, because more energy need to be taken to collect data. By contrast, FF\_UFS performs better than approAlgNoNei. More specifically, for every minute increase in the mean value of service time, the number of flights required by FF\_UFS increases between 3.75 and 7.6, while that of approAlgNoNei increases between 6.35 and 10.35. It indicates that the gap between the results of FF\_UFS and approAlgNoNei is related to the mean value of GCs' service time.

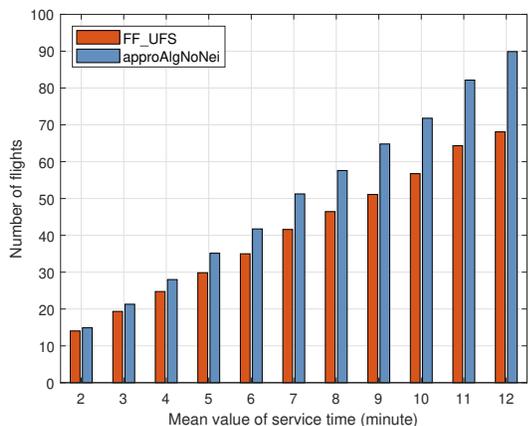


Fig. 7. Performance of FF\_UFS and approAlgNoNei varying GCs' service time.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we consider data collection scenarios with a set of GCs deployed along a road, and assign multiple UAV flights to fly over GCs and receive video data by 60 GHz communication in flying and hovering mode. For such scenarios, we study the UFS problem which aims to

minimize the number of UAV flights while satisfying the data requirements of all GCs with the limitations of energy and data storage. We prove that the UFS problem is NP-hard and design efficient algorithms with theoretical approximation ratios. Specifically, we first study the special case of the UFS problem where all the cameras are in the same direction of the ground station, and propose algorithms NF\_SUFS and FF\_SUFS, whose approximation ratios are both proved to be 2. Then, we extend the algorithms to a more general case with the ground station located at an arbitrary position, and put forward the FF\_UFS algorithm that achieves an approximation ratio of 3. Finally, we conduct experiments to validate the effectiveness and efficiency of our algorithms.

In the future, we plan to investigate the UAV flight scheduling problem for data collection with time windows, that is, the data transmission of each GC must be completed within its own time window. Moreover, we will further consider data freshness while optimizing the scheduling of UAV flights for data collection.

## ACKNOWLEDGMENT

This work was partially supported by the National Natural Science Foundation of China (Nos. 62072102, 62132009, 62072101, and 62072103); Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201); the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China (No. 93K-9); the Fundamental Research Funds for the Central Universities (No. 2242022k30029).

## REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.
- [2] B. Ahlgren, M. Hidell, and E. C. H. Ngai, "Internet of Things for smart cities: Interoperability and open data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, Nov 2016.
- [3] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An IoT platform for data-driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, Boston, MA, Mar. 2017.
- [4] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.
- [5] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, and G. Chen, "Millimeter wave communication: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018.
- [6] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5G: opportunities and challenges," *Wireless networks*, vol. 21, no. 8, pp. 2657–2676, 2015.

- [7] A. N. Uwaechia and N. M. Mahyuddin, "A comprehensive survey on millimeter wave communications for fifth-generation wireless networks: Feasibility and challenges," *IEEE Access*, vol. 8, pp. 62 367–62 414, 2020.
- [8] A. Ghosh, O. D. Incel, V. S. A. Kumar, and B. Krishnamachari, "Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, Macau, China, Oct 2009, pp. 363–372.
- [9] L. Sudha and P. Thangaraj, "Improving energy utilization using multi hop data aggregation with node switching in wireless sensor network," *Cluster Computing*, vol. 22, no. 5, pp. 12 749–12 757, 2019.
- [10] S. K. Saha, H. Assasa, A. Loch, N. M. Prakash, R. Shyamsunder, S. Aggarwal, D. Steinmetzer, D. Koutsonikolas, J. Widmer, and M. Hollick, "Fast and infuriating: Performance and pitfalls of 60 GHz WLANs based on consumer-grade hardware," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.
- [11] H. Assasa, S. K. Saha, A. Loch, D. Koutsonikolas, and J. Widmer, "Medium access and transport protocol aspects in practical 802.11 ad networks," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018, pp. 1–11.
- [12] E. Udayakumar and V. Krishnaveni, "Analysis of various interference in millimeter-wave communication systems: A survey," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–5.
- [13] Y. Gu, F. Ren, Y. Ji, and J. Li, "The evolution of sink mobility management in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 507–524, 2015.
- [14] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.
- [15] X. Sun, D. W. K. Ng, Z. Ding, Y. Xu, and Z. Zhong, "Physical layer security in UAV systems: Challenges and opportunities," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 40–47, 2019.
- [16] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: Challenges, solutions, and comparisons," *Computer Communications*, vol. 151, pp. 518–538, 2020.
- [17] T. Shen and H. Ochiai, "A UAV-enabled wireless powered sensor network based on NOMA and cooperative relaying with altitude optimization," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 21–34, 2021.
- [18] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "Data communication in linear wireless sensor networks using unmanned aerial vehicles," in *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 492–499.
- [19] X. Ren, W. Liang, and W. Xu, "Use of a mobile sink for maximizing data collection in energy harvesting sensor networks," in *International Conference on Parallel Processing*, 2013.
- [20] J. Gong, T. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1942–1954, Sep. 2018.
- [21] R. Vishnuvarthan, R. Sakthivel, V. Bhanumathi, and K. Muralitharan, "Energy-efficient data collection in strip-based wireless sensor networks with optimal speed mobile data collectors," *Computer Networks*, vol. 156, pp. 33 – 40, 2019.
- [22] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86–99, Jan 2012.
- [23] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Information Processing in Sensor Networks*, Berlin, Heidelberg, 2003, pp. 129–145.
- [24] E. Ekici, Y. Gu, and D. Bozdog, "Mobility-based communication in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 7, pp. 56–62, 2006.
- [25] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, pp. 7:1–7:31, Aug. 2011.
- [26] H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Computer Networks*, vol. 148, pp. 1 – 19, 2019.
- [27] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 592–608, April 2011.
- [28] A. E. A. A. Abdulla, Z. M. Fadlullah, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "An optimal data collection technique for improved utility in UAS-aided networks," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, April 2014, pp. 736–744.
- [29] E. Tuyishimire, A. Bagula, S. Rekhis, and N. Boudriga, "Cooperative data muling from ground sensors to base stations using UAVs," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 35–41.
- [30] Q. Zhang, W. Xu, W. Liang, J. Peng, T. Liu, and T. Wang, "An improved algorithm for dispatching the minimum number of electric charging vehicles for wireless sensor networks," *Wireless Networks*, vol. 25, no. 3, pp. 1371–1384, Apr 2019.
- [31] R. A. Nazib and S. Moh, "Energy-efficient and fast data collection in UAV-aided wireless sensor networks for hilly terrains," *IEEE Access*, vol. 9, pp. 23 168–23 190, 2021.
- [32] J. Zhang, Z. Li, W. Xu, J. Peng, and X. Jia, "Minimizing the number of deployed uavs for delay-bounded data collection of iot devices," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021.
- [33] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, 2018.
- [34] S. Basagni, G. Koutsandria, and C. Petrioli, "Enabling the mobile IoT: Wake-up unmanned aerial systems for long-lived data collection," in *Proceedings of IEEE MASS 2019*, Monterey, CA, November 4–7 2019, pp. 1–8.
- [35] D. Ebrahimi, S. Sharafeddine, P. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1893–1905, April 2019.
- [36] L. Tao, X. M. Zhang, and W. Liang, "Efficient algorithms for mobile sink aided data collection from dedicated and virtual aggregation nodes in energy harvesting wireless sensor networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 4, pp. 1058–1071, 2019.
- [37] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, June 2018.
- [38] Y. Fan, Q. Zhang, and L. Gui, "Exploiting an unmanned aerial vehicle for energy efficient data collection in wireless sensor networks," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, Chengdu, China, Dec 2018, pp. 939–943.
- [39] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.
- [40] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, Oct 2019.
- [41] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.
- [42] S. Gautam and T. R. Rao, "Beamforming investigation for wireless communication at 60 GHz," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 1765–1768.
- [43] B. Satchidanandan, S. Yau, P. Kumar, A. Aziz, A. Ekbal, and N. Kundargi, "TrackMAC: An IEEE 802.11 ad-compatible beam tracking-based mac protocol for 5G millimeter-wave local area networks," in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2018, pp. 185–182.
- [44] Y.-Y. Li, C.-Y. Li, W.-H. Chen, C.-J. Yeh, and K. Wang, "Enabling seamless wigo/wifi handovers in tri-band wireless systems," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–2.
- [45] A. Scholl, R. Klein, and C. Jurgens, "Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem," *Computers & Operations Research*, vol. 24, no. 7, pp. 627–645, 1997.
- [46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
- [47] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: <http://www.gurobi.com>



**Wenjia Wu** received the B.S. and Ph.D. degrees in computer science in 2006 and 2013, respectively, from Southeast University, China. He is an associate professor at the School of Computer Science and Engineering in Southeast University. His research interests include wireless and mobile networks.



**Shengyu Sun** received the B.E. degree from Hohai University, Nanjing, China, in 2019. She is currently a M.S. student in the School of Computer Science and Engineering in Southeast University, Nanjing, China. Her research interests include wireless and IoT networks.



**Feng Shan** received the Ph.D. degree in computer science from Southeast University, Nanjing, China, in 2015. He was a visiting student with the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, wireless networks, swarm intelligence, and algorithm design and analysis.



**Ming Yang** received the M.S. and Ph.D. degrees in computer science and engineering at Southeast University, China, in 2002 and 2007, respectively. He is currently a professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include network security, privacy and Internet of Things.



**Junzhou Luo** received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in computer network, all from Southeast University, China, in 1982, 1992, and 2000, respectively. He is a full professor in the School of Computer Science and Engineering, Southeast University. He is a member of the IEEE Computer Society and co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design, and he is a member of the ACM and chair of ACM SIGCOMM China. His research interests are next generation network

architecture, network security, cloud computing, and wireless LAN.